PSN College of Engineering and Technology (Autonomous)

Tirunelveli – 627152

COURSE PLAN

Academic year (2023-24)

(Regulation- 2018)

Subject Name	Software Engineering
Course Type	Core Paper
Programme	B. E , Computer Science and Engineering
Year/ Semester/ Section	II / IV
Nature of Course / Credit	Theory / 3
Course Coordinator	
Course Code	

Course File Verification and Auditing

Part-I
(At the beginning of the semester)

Submission Date	Check List	Verified by HOD	Verified by Academic Auditor

	Vision and Mission	Course Description, Objective and Outcomes	CO-PO mapping	Course Plan and Target	Syllabus and Content beyond Syllabus	Assignments & additional resources	Course Delivery Plan	University Question Papers		

Part-II (After CAT - I)

Submission Date	Check List					Verified by HOD	Verified by Academic Auditor
	Syllabus Coverage	Notes and Other Materials	Performance Analysis	Feedback	Proof for Participatory Learning		

Part-III (After CAT - II)

Submission Date	Check List					Verified by HOD	Verified by Academic Auditor
	Syllabus Coverage	Notes and Other Materials	Performance Analysis	Question Papers and Keys	Proof for Participatory Learning		

Part-IV (After Model examination)

Submission Date	Check List					Verified by HOD	Verified by Academic Auditor
	Syllabus Coverage	Notes and Other Materials	Performance Analysis	Question Papers and Keys	Proof for Participatory Learning		

Semester Academic Audit

Audit Remarks:
Signature of the Auditor(s):
Signature of Director (Academics) Signature of Principal

Contents

6	Instructional Learning outcomes	
7	Program Educational Objectives (PEOs)	
8	Program Outcomes (POs)	
9	Program Specific Outcomes (PSOs)	
10	CO and PO mapping	
11	Text Books & Reference Books	
12	Web resources	
13	E – learning links	
14	Magazines & Journals	
15	Lesson Plan	
16	Class time table	
17	Course time table	
18	Content Delivery Methodologies	
19	Assignments	
20	Assignment Rubrics	
21	Mapping of CO to Assignment	
22	Assessment Methodologies	
23	Distribution of portions for assessment tests	
24	Mark Allotment for CO Assessment	
25	Lecture Notes	
26	Content beyond syllabus	
27	Question Bank	
28	End Semester questions papers	
29	Students' Name list	
30	Identification of fast and slow learners	
31	Remedial Action for slow learners	
32	Encouragements for fast learners	
33	Students' group list for topic discussion	
34	Course Review & Closure Report	

VISION AND MISSION OF THE INSTITUTE:

Institution Vision	Emerge as a pioneer institute inculcating engineering education and skills, research, values and ethics.

Institution Mission	**IM-1**	To achieve greater heights of excellence in technical knowledge and skill development through innovative teaching and learning practices.
	IM-2	To develop the state of art infrastructure to meet the demands of technological revolution.
	IM-3	To improve and foster research in all dimensions for betterment of society.
	IM-4	To develop individual competencies to enhance innovation, employability and entrepreneurship among students.
	IM-5	To instill higher standards of discipline among students, inculcating ethical and moral values for societal harmony and peace.

VISION AND MISSION OF THE DEPARTMENT:

Department Vision	To emerge as a preeminence program to produce quality Computer Science and Engineering graduates.	
Department Mission	**DM-1**	To enhance professional and entrepreneurial skills through industry institute interaction to enable them in getting better placement.
	DM-2	To promote research and continuing education.
	DM-3	To train the students according to their discipline to meet dynamic needs of the society.

1. PRE REQUISITES

NIL

2. COURSE DESCRIPTIONS

Software engineering is a discipline that allows us to apply engineering and computer science concepts in the development and maintenance of reliable, usable, and dependable software. The course is designed to present software engineering concepts and principles in parallel with the software development life cycle. The course is designed to reflect the phases like analysis, design, and testing and software metrics as followed in the software company.

3. CAREER OPPORTUNITIES:

- Software Engineer
- Principal Software Engineer
- Programmer Analyst
- Application Developer
- Lead Software Development Engineer
- Software architect
- Software project manager
- Test engineer
- Software developer

4. SYLLABUS

5030009	Software Engineering	L	T	P	C
		3	0	0	3

Unit – I Software Product and Process 9

Introduction – Software Engineering Layers – Software Process – Generic Process Model –Prescriptive Process Model: Water Fall Model – Increment Process Model – Prototyping Model – Spiral Model – Concurrent Model. Specialized Process Model: Component Based Development - Formal Methods Model – Aspect Oriented Software Development.

Unit – II Requirement Engineering and Analysis 9

Requirement Engineering Tasks – building requirement Model – Requirement Analysis: Objective(s)s, Rules, Domain Analysis, Requirement modeling Approach, Scenario Based Modeling: Creating, Refining, Writing a Use case. UML Models: Activity Diagram, Swim lane Diagrams. Data Modeling – class Based Modeling.

Unit – III Design Concepts and Principles 9

Design within the context of Software Engineering - Design Process - Design Concepts – The Design Model – Architecture Model- Architecture Design – Architecture Mapping using Data flow.

Unit – IV Testing 9

Strategic approach to Software Testing: Verification and Validation. Organizing for software Testing – Software Testing Strategy – Criteria for Completion of Testing. Unit Testing - Integration Testing – Regression Testing – Smoke Testing – Validation Testing – System Testing – The Art of Debugging.

Unit – V Software Project Management 9

Project Management Concepts: Management Spectrum – People – Product – Process – Project. Process and Project Metris – Process Metrics & Software Process improvement – Project Metrics. Software Measurement: Size Oriented Metrics – Function Oriented Metrics – Use case oriented metrics- WebApp Project Metrics–Empirical Estimation model- COCOMO II Model- Risk Management

Total: 45 Periods

Text Book:

1. Roger S. Pressman, "Software Engineering – A practitioner's Approach", Seventh Edition, McGraw-Hill International Edition, 2010(UNIT I-V)

Reference Book(s):

1. Watts S.Humphrey, "A Discipline for Software Engineering", Pearson Education, 2007.
2. James F.Peters and Witold Pedrycz, "Software Engineering, an Engineering Approach", Wiley-India, 2007.
3. Stephen R.Schach, "Software Engineering", Tata McGraw-Hill Publishing Company Limited, 2007.
4. S.A.Kelkar, "Software Engineering", Prentice Hall of India Pvt., 2007.

5. COURSE OUT COMES

After successful completion of the course, the students should be able to		
CO's	**CO – STATEMENTS**	**PO's**
CO1	To Understand the fundamental concepts of Generic and Specialized process model	1,3
CO2	Understand the concept of Analysis and Modeling activity	2,4,5,10
CO3	Analyze the concepts of software design	3,4,10
CO4	Understand the different software testing methods	4,5
CO5	Understand the concept of software process metrics and measurement	11

6. INSTRUCTIONAL LEARNING OUTCOMES

UNIT	LEARNING OUTCOMES
I	The outcome will be assess through assignment-1, tutorial-1, Class test -1, MCQ Test-1, CAT-1.
II	The outcome will be assess through assignment-2, tutorial-2, Class test -2, MCQ Test-2, CAT-1, CAT – 1&2.

III	The outcome will be assess through assignment-3, tutorial-3, Class test -3, MCQ Test-3, CAT-2.
IV	The outcome will be assess through assignment-4, tutorial-4, Class test -4, MCQ Test-4, CAT-3.
V	The outcome will be assess through assignment-5, tutorial-5, Class test -5, MCQTest-4, CAT-3. Test-5.

7. PROGRAMME EDUCATIONAL OBJECTIVES (PEO's)

S. No	Topic	PEOs
PEO1	**Fundamental Knowledge**	Graduates will be able to perform in technical and managerial roles ranging from design, development and problem solving to suit to the industrial needs
PEO2	**Career Development**	Graduates will be able to successfully pursue higher education and also Graduates will have the ability to adapt, contribute and innovate new technologies in different domains of computer science & Engineering
PEO3	**Social Identity**	Graduates will be ethically and socially responsible engineers in computer science & Engineering disciplines

8. PROGRAM OUTCOMES [PO's]

PO'S NO	KNOWLEDGE	STATEMENTS	APPLIANCE
1	Engineering Knowledge:	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	Theory/ Practical / Project work
2	Problem Analysis:	Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	Theory / Practical / Projects
3	Design / Development of Solutions:	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	Theory / Practical / Projects

4	Conduct Investigations of Complex Problems:	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	Theory / Practical
5	Modern Tool usage:	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	Theory / Practical / Project work
6	The Engineer and Society:	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	Theory / Industrial visit / In plant training
7	Environment and Sustainability:	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	Theory / Industrial Visit/ In plant Training
8	Ethics:	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	Theory / Industrial visit / In plant Training
9	Individual and Team Work:	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	Projects
10	Communication:	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	Projects/ Seminar/ Mini Project
11	Project Management and Finance:	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	Projects
12	Life-long Learning:	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	Projects / Higher Studies

8. PROGRAMME SPECIFIC OBJECTIVE (PSO's)

PSO1	The computer science and Engineering graduates are able to analyze, design, develop, test and apply Management strategy, Mathematical concept in the development of computational solutions, make them expert in computer software and hardware.
PSO2	Develop their skills to solve problems in all area of programming concepts and adopt high environmental and social issues with ethics to manage different projects in inter-disciplinary field.

9. CO- PO MAPPING

CO's NO	COURSE OUTCOME	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	To Understand the fundamental concepts of Generic and Specialized process model	2	-	2	-	-	-	-	-	-	-	-	-	2	-
CO2	Understand the concept of Analysis and Modeling activity	-	3	-	2	2	-	-	-	-	2	-	-	-	2
CO3	Analyze the concepts of software design	-	-	3	2	-	-	-	-	-	2	-	-	-	2
CO4	Understand the different software testing methods	-	-	-	3	2	-	-	-	-	-	-	-	-	2
CO5	Understand the concept of software process metrics and measurement	-	-	-	-	-	-	-	-	-	-	3	-	-	2

11. TEXT BOOK & REFERENCE BOOK LIST

Sl. No	Description	Legend
Text Book(s):		
1	Roger S. Pressman, "Software Engineering – A practitioner's Approach", Seventh Edition, McGraw-Hill International Edition, 2010	T1
Reference Book(s):		
1	Watts S.Humphrey, "A Discipline for Software Engineering", Pearson Education, 2007.	R1
2	James F.Peters and Witold Pedrycz, "Software Engineering, an Engineering Approach", Wiley-India, 2007.	R2
3	Stephen R.Schach, "Software Engineering", Tata McGraw-Hill Publishing Company Limited, 2007.	R3
4	S.A.Kelkar, "Software Engineering", Prentice Hall of India Pvt., 2007.	R4

12. Web Resources

Sl. No	Topic	Web link
1	Software testing	https://www.tutorialspoint.com/software_testing/index.htm

13. E- learning / NPTEL

NPTEL/ OTHER UNIVERSITY video lectures related to syllabus:	
V1	https://nptel.ac.in/courses/106/105/106105182/
LN1	https://web.cs.dal.ca/~hawkey/3130/SEBackground4.pdf

14. MAGAZINE & JOURNALS

Magazine	Software Development News- SD Times Software Magazines IEEE
Journals	***Information Systems Journal*** ***Business and Information Systems Engineering*** ***Empirical Software Engineering*** ***IEEE Transactions on Software Engineering*** ***Advances in Engineering Software*** ***IEEE Transactions on Reliability***

15. Lesson Plan & Content Delivery Methodologies

S.No.	Unit	Topic to be covered	Hours Needed	Mode of Teaching (BB/PPT/Others)	Text/ Ref. Book	Page No.
Software Product and Process						
1	I	Introduction	1	**BB**	**T1**	**15**
2		Software Engineering Layers	1	**BB**	**T1**	**20**
3		Process – Generic Process Model	1	**BB**	**T1**	**29**
4		Prescriptive Process Model: Water Fall Model	1	**BB**	**T1**	**39**
5		Increment Process Model	1	**BB**	**T1**	**41**
6		Prototyping Model, Concurrent Model.	1	**BB**	**T1**	**42,48**
7		Spiral Model	1	**BB**	**T1**	
8		Specialized Process Model: Component Based Development	1	**BB**	**T1**	**50**
9		Formal Methods Model – Aspect Oriented Software Development	1	**BB**	**T1**	**51-52**
10		**Win Win Spiral model*	1	**BB**	-	-
Requirement Engineering and Analysis						
11	II	Requirement Engineering Tasks – building requirement Model	1	**BB**	**T1**	**120**
12		Analysis: Objective(s)s, Rules	1	**BB**	**T1**	**128**
13		Domain Analysis, Requirement modeling Approach	1	**BB**	**T1**	**129**
14		Scenario Based Modeling: Creating, Refining	1	**BB**	**T1**	**132**
15		Writing a Use case	1	**BB**	**T1**	**133**
16		UML Models	1	**BB**	**T1**	**134**
17		Activity Diagram	1	**BB**	**T1**	**161**
18		Swim lane Diagrams	1	**BB**	**T1,R1**	**162**
19		Data Modeling – class Based Modeling	1	**BB**	**T1,R1**	**167**
Design Concepts and Principles						
20	III	Introduction to design phase	1	**BB**	**T1**	**215**
21		Design within the context of Software Engineering	1	**BB**	**T1**	**216**
22		Design Process	1	**BB**	**T1**	**219**
23		Design Concepts	1	**BB**	**T1**	**222**
24		The Design Model	2	**BB**	**T1**	**233**
25		Architecture Model	1	**BB**	**T1**	**243**
26		Architecture Design	1	**BB**	**T1**	**255**
27		Architecture Mapping using Data flow.	1	**BB**	**T1**	**265**
28		* Data design and data warehousing	1	**BB**	**T1**	-
Testing						
29	IV	Strategic approach to Software Testing	1	**BB**	**T1,R2**	**449**

S.No.	Unit	Topic to be covered	Hours Needed	Mode of Teaching (BB/PPT/Others)	Text/ Ref. Book	Page No.
30		Verification and Validation	1	**BB**	**T1**	**450**
31		Organizing for software Testing	1	**BB**	**T1**	**451**
32		Software Testing Strategy – Criteria for Completion of Testing	1	**BB**	**T1,R2**	**452**
33		Unit Testing	1	**BB**	**T1.R2**	**456**
34		Integration Testing	1	**BB**	**T1**	**459**
35		Regression Testing	1	**BB**	**T1**	**466**
36		Smoke Testing – Validation Testing	1	**BB**	T1	**467**
37		System Testing – The Art of Debugging	1	**BB**	**T1**	**470-475**
38		* Automation testing tools	2	**BB**	**-**	**-**
Software Project Management						
39	**V**	Project Management Concepts: Management Spectrum	1	**BB**	**T1,R3**	**646**
40		People – Product	1	**BB**	**T1,R3**	**649-656**
41		Process – Project.	1	**BB**	**T1**	**657-660**
42		Metris – Process Metrics & Software Process improvement	1	**BB**	**T1**	**667**
43		Project Metrics. Software Measurement: Size Oriented Metrics	1	**BB**	**T1**	**670**
44		– Function Oriented Metrics – Object Oriented Metrics	1	**BB**	**T1**	**673-675**
45		Use case Oriented Metrics. Web Application-Project Metrics	1	**BB**	**T1**	**676,677**
46		Empirical Estimation model	1	**BB**	**T1,R4**	**708**
47		COCOMO II Model	1	**BB**	**T1,R4**	**709**
48		Risk Management	1	**BB**	**T1,R4**	**744**
Total Hours Needed: 45(L) + 3* = 48 Hours						

***Content beyond Syllabus**

16. CLASS TIME TABLE

17. COURSE TIME TABLE

TIME TABLE [2022-23 - ODD SEMESTER]

Year : **II CSE** **Semester** : **04**

Revision No : **0** **Effective Date** : **8.11.22**

Hour/Day	**9.00 – 10.00**	**10.00 – 11.00**	**11.00 – 11.10**	**11.10 – 12.10**	**12.10 – 1.10**	**1.10 – 2.00**	**2.00 – 2.50**	**2.50 – 3.40**	**3.40 – 3.50**	**3.50 - 4.40**
MON				SE		**LUNCH BREAK**			**BREAK**	
TUE					SE					
WED					SE					
THUR										
FRI	**Test** SE									

19. Assignments

Sl.No	Topic	Date of Submission
1.	Levels of Capability Maturity Model (CMM)	31-01-2023
2.	Activity and Swim lane diagram for Library Management application	16-02-2023
3.	Transform and Transaction Mapping	01-03-2023
4.	Black box and White box testing	14-03-2023
5.	Function point analysis for Safe home architecture	23-03-2023

20. Assignment Rubrics

Quality	Marks
Submission on Date	2
Understanding	3
Solving Skills/Presentation	3
End results with correct units conversions / Conclusion	2
TOTAL	10

21. MAPPING COs with ASSIGNMENTS

CO's	CO – STATEMENTS	A1	A2	A3	A4	A5
CO1	To Understand the fundamental concepts of Generic and Specialized process model	3	-	-	-	-
CO.2	Understand the concept of Analysis and Modeling activity	-	3	-	-	-
CO3	Analyze the concepts of software design	-	-	3	-	-
CO4	Understand the different software testing methods	-	-	-	3	-
CO5	Understand the concept of software process metrics and measurement	-	-	-	-	3

22. ASSESSMENT METHODOLOGIES

Assessment Tool			Description
Direct Assessment (80%)	Internal Test	20%	25% each for CAT – I, CAT – II, Model Examination and Assignments.
	Assignments		
	Practical lab (Internal)		-
	End semester Examination	80%	100% for End Semester Examination
	Practical lab (External)		-
Indirect Assessment (20%)	Course End Survey (80%)		Enclosed separately
	Exit Survey (20 %)		At the end of the course completion will be evaluated

23. DISTRIBUTION OF PORTIONS FOR ASSESSMENT TESTS

Assessments	Portion Covering Unit	% of weightage
CAT - I	Unit-1 and 2	40
CAT - II	Unit-3 and 4	40
Model Examinations	Unit - 1 to 5	100
Assignments – 1	Unit - 1	20
Assignments - 2	Unit - 2	20
Assignments - 3	Unit -3	20
Assignments - 4	Unit - 4	20
Assignments - 5	Unit - 5	20
End Semester	Unit - 1 to 5	100

24. Mark Allotment for CO Assessment:

COs	CAT - I	CAT - II	Model Examination	Assignment	MCQ	End Semester
CO1	30		20	20	20	20
CO2	30		20	20	20	20
CO3		30	20	20	20	20

CO4		30	20	20	20	20
CO5			20	20	20	20

25. LECTURE NOTES: Enclosed separately

26. CONTENT BEYOND SYLLABUS

UNIT	TOPICS TO BE COVERED	Hrs Taken
1- Topic	**Software Product and Process**	
BEYOND SYLLABUS	Win Win Spiral model	**1**
3.Topic	**Design Concepts and Principles**	
BEYOND SYLLABUS	Data design and data warehousing	**1**
4.Topic	Testing	
BEYOND SYLLABUS	Automation testing tools	**1**

27. Question Bank

SUBJECT : Software Engineering

Sub Code :

Semester : 4

Year : 2019-20

Batch : 2018– 2022

Prepared by :
Dr.G.NALLASIVAN, M.E.,Ph.D
Professor/CSE
PSNCET

PSN COLLEGE OF ENGINEERING AND TECHNOLOGY (Autonomous)
Melathediyoor – 627 152
DEPARTMENT OF CSE & IT
II year IV semester CSE and IT
QUESTION BANK- SOFTWARE ENGINEERING

=S.No.	Question	CO	BT	Usage*
	UNIT I			
	PART –A (TWO MARKS)			
1	How would you define software process? *Software Process* is a set of required activities and the outcome of the activities with a target to produce a software product	**CO1**	**L-1**	
2	Outline the characteristics of software . Software characteristics are classified into 6 major components: Functionality, Reliability, Efficiency, Usability:, Efficiency and Maintainability	**CO1**	**L-II**	
3	Define Umbrella activity Umbrella Activities are those activities to be performed through the entire Software Process. Eg; SCM, Quality assurance etc	**CO1**	**L-1**	
4	How would you define CMM? The Capability Maturity Model (CMM) is a methodology used to develop and refine an organization's software development process.. CMM was developed and is promoted by the Software Engineering Institute (SEI), a research and development center sponsored by the U.S. Department of Defense (DoD).	**CO1**	**L-1**	
5	Examine the drawbacks of Waterfall model. The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.	**CO1**	**L-IV**	
6	How would you outline the advantages of Incremental process model? ● Generates working software quickly and early during the software life cycle. ● More flexible – less costly to change scope and requirements. ● Easier to test and debug during a smaller iteration. ● Easier to manage risk because risky pieces are identified and handled during its iteration	**CO1**	**L-I**	
7	How would you outline the advantages of prototyping model? ● Users are actively involved in the development. ● Since in this methodology a working **model** of the system is provided, the users get a better understanding of the system being developed. ● Errors can be detected much earlier. ● Quicker user feedback is available leading to better solutions.	**CO1**	**L-I**	
8	What can you infer from evolutionary process model?	**CO1**	**L-I**	

	Evolutionary model is a combination of Iterative and Incremental model of software development life cycle. Delivering your system in a big bang release, delivering it in incremental process over time is the action done in this model.			
9	What are the advantages of spiral model? High amount of risk analysis hence, avoidance of Risk is enhanced. Good for large and mission-critical projects. Strong approval and documentation control. Additional Functionality can be added at a later date. Software is produced early in the software life cycle.	**CO1**	**L-I**	
9	Examine the advantages of Formal method model? There are a number of advantages to the Formal Methods Model. They are as follows: Uncovers problems quickly because the specification language syntax enforces correctness. Promotes problem-free software because each step is verified/validated along the way.	**CO1**	**L-IV**	
10	How would you outline the features of aspect oriented software model? Considered a subset of post-object programming technologies. Better software design support through isolating application business logic from supporting and secondary functions. Provides complementary benefits and may be used with other agile processes and coding standards.	**CO1**	**L-I**	
11	How do you define stakeholders? The Person who is having direct or indirect influence on the success of the product called stakeholders. Eg. Developer, Analyst, Design Engineer, Trade union Representatives, Domain Experts etc	**CO1**	**L-I**	
	UNIT 2			
12	What are the types of requirements? Functional Non-functional User System Domain	**CO2**	**L-I**	
13	State the advantages of Requirement elicitation and analysis. This technique provides a complete set of requirements as compared to brainstorming and group meetings. Requirements are mostly unchanged whenever elicited with this type of method. This technique is feasible for the large and complex solution.	**CO2**	**L-I**	
14	What is meant by Domain analysis? Domain analysis is the process of identifying, collecting, organizing, analyzing and representing a domain model from the study of existing systems, underlying theory, emerging technology and development histories within the domain of interest.	**CO2**	**L-I**	
15	What do you mean by Inception? Inception refers more to the beginning, to entering upon an undertaking. Inception implies the start of a specific thing like a campaign or a company. Subsequent events take place after the inception. At the	**CO2**	**L-I**	

	moment of conception, most women are at the inception of motherhood. Definitions of inception.			
16	What is the various Modeling approach? Use case Model Scenario based Model Data-Class based Model Activilty Swimlane diagram	**CO2**	**L-I**	
17	What is meant by UML diagram? A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.	**CO2**	**L-I**	
18	Justify Scenario based Modeling is suitable for all applications. **Scenario-based modeling** is one of the sub-stages of requirements **modeling**. It is also typically the first stage of requirements **modeling**, as it identifies the primary use cases for the proposed software system or application, to which later stages of requirements **modeling** will refer.	**CO2**	**L-V**	
19	What is the advantage of Scenario based Modeling? Using the **scenario** planning , companies can predict the possible future **scenarios**, so that managers can be more aware of the external environment and their company's status and future directions therefore they can match internal resources to external environment to allocate the resources more effective and rational	**CO2**	**L-I**	
20	Show the importance of class based Modeling. *Class-based Modeling* In the context of software engineering, requirements *modeling* examines the requirements a proposed software application or system must meet in order to be successful. ... *Class-based modeling* takes the use case and extracts from it the *classes*, attributes, and operations the application will use.	**CO2**	**L-II**	
21	Demonstrate the features of Generalization in class diagram? Generalization is the process of extracting shared characteristics from two or more classes, and combining them into a generalized superclass. Shared characteristics can be attributes,	**CO2**	**L-II**	
22	Assess the use of Extend in class diagram? extend in use case diagram? Extend is used when a use case conditionally adds steps to another first class use case.	**CO2**	**L-V**	
	UNIT 3			
23	State any two Quality criteria guidelines. Correctness: User review is used to ensure **the** correctness of **requirements** stated in **the** SRS.	**CO3**	**L-I**	

	▪ Completeness ▪ Consistency			
24	Examine the purpose of design. Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. . The output of this process can directly be used into implementation in programming languages.	**CO3**	**L-IV**	
25	What is meant by abstraction? **Abstraction** is the act of representing essential features without including the background details or explanations. In the computer science and software engineering domain, the **abstraction** principle is used to reduce complexity and allow efficient design and implementation of complex software systems.	**CO3**	**L-I**	
26	What is Refinement? **Refinement** is a generic term of computer science that encompasses various approaches for producing correct computer programs and simplifying existing programs to enable their formal verification.	**CO3**	**L-I**	
27	Analyze the use of Modularity. The concept of **modularity** is used primarily to reduce complexity by breaking a system into varying degrees of interdependence and independence across and "hide the complexity of each part behind an abstraction and interface	**CO3**	**L-IV**	
28	What is meant by Refactoring? Refactoring is "the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure," according to Martin Fowler, the "father" of refactoring.	**CO3**	**L-I**	
29	Justify functional independence is suitable for improving Modularity. Functional independence is the ability to carry out activities of daily living safely and autonomously. Data Element Concept: Person—level of functional independence.	**CO3**	**L-V**	
30	What is coupling? Coupling is the degree of interdependence between software modules; a measure of how closely connected two routines or modules are; the strength of the relationships between modules.	**CO3**	**L-I**	
31	What is cohesion? Cohesion refers to the level of strength and unity with which different components of a software program are inter-related with each other.	**CO3**	**L-I**	
32	Compare and contrast about Fan out and Fan in. **Fan**-in is **defined** as the number of in inputs a gate has. For example A two input gate will have fin-in = 2. **Fan-out**:- it is the maximum number of inputs of the same IC family that a gate can Drive without falling outside the specified output voltage limits	**CO3**	**L-II**	
33	Outline about component level design.	**CO3**	**L-II**	

	Component-level design occurs at the stage in software engineering following the architectural **design** of the software. It translates that **design** into a more detailed **design** before the actual coding of the software begins. A **component** is the basic building block of the software.			
34	What is meant by interface design? User **interface engineering** is the **design** of user **interfaces** for machines and **software**, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience.	**CO3**	**L-I**	
35	What is meant by pipe and filter pattern of architectural design? The Pipe and Filter is an architectural design pattern that allows for stream/asynchronous processing. In this pattern, there are many components, which are referred to as filters, and connectors between the filters that are called pipes.	**CO3**	**L-I**	
36	Summarize the features about Transform flow. Information enter at a point, transformed, and exit at another point. Path: incoming flow central→ transform outgoing→ flow. Transaction Flow. Single data item, transaction, triggers one of many paths.	**CO3**	**L-II**	
37	What is meant by transaction flow? A **transaction** consists of a sequence of operations, some of which are performed by a system, persons or devices that are outside of the system. **Transaction** begin with Birth-that is they are created as a result of some external act.	**CO3**	**L-I**	
38	Define Fan in and Fan out. **Fan**-in is **defined** as the number of in inputs a gate has. For example A two input gate will have fin-in = 2. **Fan-out**:- it is the maximum number of inputs of the same IC family that a gate can Drive without falling outside the specified output voltage limits.	**CO3**	**L-I**	
	UNIT -4			
39	Discover the importance of verification and validation. Verification is the process of checking that a software achieves its goal without any bugs. Verification means **Are we building the product right?** Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. Validation means **Are we building the right product?**	**CO4**	**L-IV**	
40	What is meant by Black box testing? *Black box testing* is *defined* as a *testing* technique in which functionality of the Application Under *Test* (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software.	**CO4**	**L-I**	
41	Outline the features of functional testing. Identify **functions** that the software is expected to perform. Create input data based on the **function's** specifications. Determine the output based on the **function's** specifications. Execute the **test** case.	**CO4**	**L-II**	

	Compare the actual and expected outputs.			
42	Show the structural testing is similar to white box testing. **Structural testing** is the type of **testing** carried out to **test** the **structure** of code. It is also known as White Box **testing** or Glass Box **testing**. This type of **testing** requires knowledge of the code, so, it is mostly done by the developers.	**CO4**	**L-II**	
43	What is meant by white box testing? **White Box Testing** is defined as the **testing** of a software solution's internal structure, design, and coding. In this type of **testing**, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security.	**CO4**	**L-I**	
44	What is Unit testing? *UNIT TESTING* is a level of software *testing* where individual units/ components of software are tested. . A *unit* is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a *unit* may be an individual program, function, procedure, etc.	**CO4**	**L-I**	
45	Outline the importance of system testing. System testing is actually a series of different tests whose sole purpose is to exercise the full computer based system.System testing is important since: a) In Software Development Life Cycle the System Testing is perform as the first level of testing where the System is tested as a whole.	**CO4**	**L-II**	
46	Identify the importance of integration testing. *Integration Testing* is defined as a type of testing where software modules are integrated logically and tested as a group.	**CO4**	**L-III**	
47	Define the big bang approach of testing. **Big Bang** is an **approach** to Integration **Testing** where all or most of the units are combined together and **tested** at one go. This **approach** is taken when the **testing** team receives the entire software in a bundle.	**CO4**	**L-I**	
48	What is Smoke testing? Smoke Testing is a kind of Software Testing performed after software build to ascertain that the critical functionalities of the program are working fine.	**CO4**	**L-I**	
49	Compare and Contrast Stub and driver. Stubs and drivers are two types of test harness, which is a collection of software and test that is configured together in order to test a unit of a program by stimulating variety of conditions while constantly monitoring its outputs and behaviour.	**CO4**	**L-IV**	
50	Infer the uses of debugging. Debugging is the process of detecting and removing of existing and potential errors (also called as 'bugs') in a software code that can cause it to behave unexpectedly or crash. To prevent incorrect operation of a software or system, debugging is used to find and resolve bugs or defects.	**CO4**	**L-II**	
51	Contrast about sandwich approach in Integration testing.	**CO4**	**L-IV**	

	Sandwich Testing is the combination of bottom-up approach and top-down approach, so it uses the advantage of both bottom up approach and top down approach. ... It is also known as the Hybrid Integration Testing.			
	UNIT 5			
52	Outline about the product metrics. Product metrics are agreed-upon measurements that help product managers and marketers evaluate the success of their product.	**CO5**	**L-II**	
53	What is meant by size oriented metrics? Size oriented software metrics are derived by normalizing quality and/or productivity measures by considering the size of the software that has been produced.	**CO5**	**L-I**	
54	Categories the various methods of software measurement. Software measurements are of two categories, namely, direct measures and indirect measures. Direct measures include software processes like cost and effort applied and products like lines of code produced, execution speed, and other defects that have been reported.	**CO5**	**L-IV**	
55	Analyze the advantages of Function oriented Metrics. Function-oriented metrics focus on how much functionality software offers. But functionality cannot be measured directly. So function-oriented software metrics rely on calculating the function point (FP) — a unit of measurement that quantifies the business functionality provided by the product.	**CO5**	**L-IV**	
56	Define Risk Management *Risk management* is the process of identifying, assessing and controlling threats to an organization's capital and earnings. These threats, or *risks*, could stem from a wide variety of sources, including financial uncertainty, legal liabilities, strategic *management* errors, accidents and natural disasters.	**CO5**	**L-I**	
57	What is meant by Software Reengineering? Software Reengineering is the process of updating software without affecting its functionality. This process may be done by developing additional features on the software and adding functionalities that may or may not be required but considered to make the software experience better and more efficient.	**CO5**	**L-I**	
58	Propose any two methods of identifying Risks. Some common methods of risk identification are: brainstorming, flowchart method, SWOT analysis, risk questionnaires and risk surveys	**CO5**	**L-IV**	
59	Define the various software project Estimation methods. Expert judgment. This is probably the most common way people get a project estimation. Comparative or analogous estimation. If your current project is similar to past ones, take the data from previous work and extrapolate it to provide your estimates for the new job. Top-down. Bottom-up. Parametric model estimating.	**CO5**	**L-I**	
60	What are the methods of reducing risk?	**CO5**	**L-I**	

	The basic **methods** for **risk** management – avoidance, retention, sharing, transferring, and loss **prevention** and **reduction** – can apply to all facets of an individual's life and can pay off in the long run			
61	What is meant by Risk mitigation? *Risk mitigation* is a strategy to prepare for and lessen the effects of threats faced by a data center. Comparable to *risk reduction*, *risk mitigation* takes steps to reduce the negative effects of threats and disasters on business continuity (BC).	**CO5**	**L-I**	
	BIG QUESTIONS UNIT 1			
1	Explain Briefly about Generic process frame work model	**CO1**	**L-V**	
2	Explain in detail about Linear sequential or waterfall model with neat diagram		**L-V**	
3	Elaborate in detail about Prototyping Model		**L-IV**	
4	Discuss in detail about Incremental process model		**L-VI**	
5	Discuss in detain about Evolutionary Process model		**L-VI**	
6	Explain in detail about Spiral model with a neat diagram.		**L-V**	
7	Explain in detail about concurrent model with a neat diagram.		**L-V**	
8	Explain any two specialized process model in detail.		**L-V**	
	UNIT 2	**CO2**		
9	Discuss in detail about Requirement engineering tasks.		**L-VI**	
10	Discuss in detail about Scenario based modeling approach.		**L-VI**	
11	Explain in detail about UML modeling with a neat diagram.		**L-V**	
12	Compare and Contrast about Swim lane diagram and Activity diagram.		**L-II**	
13	Explain in detail about Class based modeling with neat diagram.		**L-V**	
	UNIT 3	**CO3**		
14	Explain in detail about various design concepts.		**L-V**	
15	Explain in detail about Design Process of design phase.		**L-V**	
16	Explain in detail about Design model with mapping from analysis phase.		**L-V**	
17	Explain about architectural style of design in detail with neat diagrams.		**L-V**	
18	Explain in detail about architectural mapping using data flow with neat diagram.	**CO3**	**L-V**	

	UNIT 4			
19	Discuss in detail about Software testing strategy with a neat diagram.		**L-VI**	
20	Discuss in detail about verification and validation with a diagram		**L-VI**	
21	Explain in detail about debugging approach for error tracing.	**CO4**	**L-V**	
22	Examine in detail about Strategic approach for software testing.		**L-IV**	
23	Explain in detail about integration testing with neat diagram.		**L-V**	
	UNIT 5			
24	Explain in detail about various software measurement approaches.		**L-V**	
25	Explain in detail about COCOMO II model.		**L-V**	
26	Compose in detail about Risk Management methods with a neat diagram.		**L-VI**	
27	Discuss in detail about software maintenance and Reengineering.	**CO5**	**L-I**	
28	Compare and compare object oriented and use case oriented metrics in detail.		**L-II**	
29	Explain briefly about Function point metrics with an example.		**L-V**	
30	Explain in detail about software process estimation.		**L-V**	

28. End Semester Question paper

Question Paper Code:

Reg. No: | | | | | | | | |

PSN COLLEGE OF ENGINEERING AND TECHNOLOGY(Autonomous)

BE DEGREE END SEMESTER EXAMINATIONS, April/ May 2018

B.E- Semester -04

203008 - Software Engineering

Regulation - 2014

Time : 3 Hours Answer ALL Questions Max. Marks 100

PART-A (10 x 2 = 20 Marks)

1. What is meant by an Umbrella activity?
2. What are the drawbacks of Waterfall model?
3. What do you mean by Validation?
4. What is Generalization in Class based modelling?
5. What is meant by Modularity?
6. What is Cohesion?
7. What is meant by stubs and drivers?
8. What do you mean by Regression testing?
9. What are the various size oriented metrics in software measurement?
10. Write about software Reengineering.

PART – B (5 x 16 = 80 Marks)

11. a) Explain any one Evolutionary software process model with a neat diagram

OR

b) Compare and explain Formal method model and aspect oriented software development model

12 a) Discuss in detail about the seven stages of requirement engineering tasks

OR

b) Compare and explain swimlane diagram and activity diagram with example

13. a) Explain briefly about various design concepts for good designing.

OR

b) Discuss in detail about the design model and architectural mapping using data flow

14 a) Explain in detail about the software testing strategies with a neat diagram.

OR

b) Explain in detail about the debugging process with a diagram

15. a) Explain in detail about the Risk management approach

OR

b) Explain in detail about the function oriented metrics with suitable example.

29. Student's Name List

CSE

S.No	Register No	Students Name
1	952221030449	Nigila Kannan
2	952221030450	NimeshVarshan S
3	952221030451	PandiSelvam J
4	952221030452	Prabhu P
5	952221030453	Prakash Rajan K
6	952221030454	Prakash S
7	952221030455	Prasanna T
8	952221030456	Rajkumar M
9	952221030457	Rakesh S
10	952221030458	Ram Kumar G
11	952221030459	Ramakrishnan N
12	952221030460	Ramalakshmi S
13	952221030461	Ravikumar C
14	952221030462	Renukudevi B
15	952221030463	Rosan
16	952221030464	Rose Mary J
17	952221030465	Sanjay Kumar R
18	952221030466	Sankar R
19	952221030467	Sathishkumar C
20	952221030468	Sathishkumar R
21	952221030469	Selvapriya R
22	952221030470	Selvaraj S

23	9522210307 1	Sembulingam S
24	9522210307 2	Senthur Pandian T
25	9522210307 3	ShanmugaPriya M
26	9522210307 4	Siva Balan T
27	9522210307 5	Sivakumar S
28	9522210307 6	Sivasankari M
29	9522210307 8	Sri Ram N
30	9522210307 9	Srivarshini R
31	9522210308 0	Suba Sangeetha S
32	9522210308 1	Subalakshmi B
33	9522210308 2	SudharshaPraba U
34	9522210308 3	Valagurunathan S
35	9522210308 4	Vignesh G
36	9522210308 5	Vignesh T
37	9522210308 6	Vishnu Priya S
38	9522210308 7	Viswajith N
39	9522210308 8	Yogalakshmi K
40	9522210330 1	Mahalingam
41	9522210330 2	Manjunath
42	9522210330 3	Mohammed Abdulla
43	9522210330 4	Nirmal kumar
44	9522210330 5	Saravanakumar
45	9522210330 6	Simson Chelladurai
46	9522210330 7	Singaravelan

30. FAST LEANER'S NAME LIST

PSN College of Engineering and Technology Department of CSE & IT			
SL. NO	**REG. NO**	**Fast Leaner's Name List**	
1	1603001	Anusha K	CSE
2	1603003	Banupriya S	
3	1603004	Booma G	
4	1603007	Gowsalya S	
5	1603008	Jeya Sheela V	
6	1603013	Radhini Rajan	
7	1603016	Rubasri K	
8	1603025	Sudha Devi T	
9	1603024	Soniya K	
10	1603031	Swetha I	
11	1603033	Vanitha M	
12	1607001	Anitha P	IT
13	1607005	Krishnaveni	
14	1607006	Ramya S	
15	1607008	Selva Lakshmi M	
16	1607009	Sivaranjini M	
17	1607010	Vasanthi S	

30. SLOW LEANER'S NAME LIST

SL. NO	REG. NO	Slow Leaner's Name List	
1	1603017	Sakthivel M	CSE
2	1603021	Selva Ganesh M	
3	1603026	Sugumaran R	
4	1603028	Suresh S	
5	1603029	Surya M	
6	1607003	Kabilarasan K	IT
7	1607007	Samuvel Abraham A	

31. REMEDIAL ACTIONS FOR SLOW LEARNERS

PSN College of Engineering and Technology
Department of CSE & IT

SL. NO	REG. NO	Remedial actions for slow learners - Coaching Classes scheduled from 4.30 pm to 5.30 pm on the following dates				
		Name	05.07.2019	03.08.2019	17.08.2019	27.08.2019
1	1603017	Sakthivel M	X	x	x	x
2	1603021	Selva Ganesh M	X	x	x	x
3	1603026	Sugumaran R	X	x	x	x
4	1603028	Suresh S	X	x	x	x
5	1603029	Surya M	X	x	x	x
6	1607003	Kabilarasan K	X	x	x	x
7	1607007	Samuvel Abraham A	X	x	x	x

32. ENCOURAGEMENT ACTIONS FOR FAST LEARNERS

PSN College of Engineering and Technology
Department of CSE & IT

SL. NO	REG. NO	Encouragement actions for fast learners	
		Name	
1	1603001	Anusha K	**The Fast learners are motivated to undergo separate Placement**
2	1603003	Banupriya S	
3	1603004	Booma G	
4	1603007	Gowsalya S	

5	1603008	Jeya Sheela V	**training program & GATE Coaching**
6	1603013	Radhini Rajan	
7	1603016	Rubasri K	
8	1603025	Sudha Devi T	
9	1603024	Soniya K	
10	1603031	Swetha I	
11	1603033	Vanitha M	
13	1607005	Krishnaveni	**The Fast learners are motivated to undergo separate Placement training program & GATE Coaching**
14	1607006	Ramya S	
15	1607008	Selva Lakshmi M	
16	1607009	Sivaranjini M	
17	1607010	Vasanthi S	

33. Group Name List for GD

PSN College of Engineering and Technology Department of CSE & IT				
SL. NO	**REG. NO**	**Name**	**Group No**	**Topic for Discussion**
1	1603001	Anusha K	G1	Security Issues in Wireless communication
2	1603003	Banupriya S		
3	1603004	Booma G		
4	1603007	Gowsalya S		
5	1603008	Jeya Sheela V		
6	1603009	Joseph Vijay M	G2	Merits of Wireless communication over Wired network
7	1603010	Lingammal K		
8	1603011	Malkiyas M		
9	1603012	Priya S		
10	1603013	Radhini Rajan		
11	1603014	Rajeshwari M	G3	Network Routing Protocols
12	1603015	Ramachandran M		
13	1603016	Rubasri K		
14	1603017	Sakthivel M		
15	1603019	Sangeetha K		
16	1603020	Saranya S	G4	Congestion control

17	1603021	Selva Ganesh M		
18	1603024	Soniya K		
19	1603025	Sudha Devi T		
20	1603026	Sugumaran R		
21	1603027	Sumathi S	G5	Functions and Features of GSM
22	1603028	Suresh S		
23	1603029	Surya M		
24	1603031	Swetha I		
25	1603032	Uma Maheswari V		
26	1603033	Vanitha M	G6	IEEE 802.11 Standards and services
27	1607001	Anitha P		
28	1607003	Kabilarasan K		
29	1607005	Krishnaveni		
30	1607006	Ramya S		
31	1607007	Samuvel Abraham A	G7	Comparision of Bluetooth, WI-FI and WI MAX
32	1607008	Selva Lakshmi M		
33	1607009	Sivaranjini M		
34	1607010	Vasanthi S		

34. Course Review & Closure Report

No. of hours prescribed for the course	45 hrs
No. of hours required or spent for the covering additional topics	5 hrs
No. of hours required to cover Assignments / Tests	5 hrs
No. of hours required for tutorials	5 hrs
No. of hours to revise the course content	5 hrs
No. of hours for any other activities related to the course	5 hrs for verifying class notes
Total hours required for course	70 hrs

Signature	Prepared by:	Approved by:		
Name :	**Dr.G.Nallasivan**	**Dr.M.Vargheese**	**Dr. X. Sahaya Shajan**	**Dr. M. Manikandan**
Designation:	Professor	HoD / CSE & IT	Dean(Academic)	Principal
Date:				